Modeling Binary Quantization via Convex Optimization Methods

1st Taiqiang Wu
Department of Electrical and Electronic Engineering
The University of Hong Kong
Hong Kong, China
takiwu@connect.hku.hk

1st ChenChen Ding
Department of Electrical and Electronic Engineering
The University of Hong Kong
Hong Kong, China
dingcc@connect.hku.hk

***Abstract*—Binary quantization, aiming to binarize weights and activations in Neural Networks into 1-bit values, greatly reduces memory usage and computing costs. Previous methods constrain the binarized values to a fixed set of learnable values. In this report, we first model the binary quantization into a convex optimization problem, which provides a better view to understand previous methods. Then we propose a two-stage Expectation Maximization style framework and a one-stage method after approximation. Experimental results demonstrate the effectiveness of the modeling process and the proposed methods.**

***Index Terms*—Binary Quantization, Convex Optimization, Expectation Maximization**

I. Introduction

Deep Neural Networks (DNNs) have shown great learning capacity in various tasks including computer vision [1], natural language processing [2], and speech processing [3]. However, heavy computing costs and memory usage make it difficult to deploy on edge devices. Therefore, model quantization [4], which aims to present the weights and activations with lower bits, has gained more and more attention.

Binary quantization, which binarizes weights and activations into 1-bit values, can effectively reduce memory usage and computing costs. For 1-bit values, we can further employ the XNOR and BitCount operations to speed up the computing process. Previous methods adapt the sign function to binarize weights and activations [5], [6]. Adabin [7] adaptively obtains the optimal binary sets $\{b_1, b_2\}$ $(b_1, b_2 \in \mathbb{R})$ of weights and activations for each layer instead of a fixed set (i.e., $\{-1, +1\}$). The key is to increase the representation ability for binarized features.

In this report, we first model the binarizing process as a convex optimization problem. The object is to reduce the gap between the output from original weights and binarized weights, under the constraint that binarized weights consist of two values. We design a novel two-stage Expectation Maximization style method to tackle the problem. Meanwhile, we employ the approximation of maximum operation to convey the problem into one differential unconstrained problem. Therefore, we can apply the traditional methods to solve this one-stage problem. Experimental results demonstrate the effectiveness of the modeling process and proposed one-stage and two-stage methods.

The main contributions of this report are as follows:

- We model the binary quantization into one convex optimization problem.
- We propose a novel two-stage Expectation Maximization style method. Meanwhile, we employ the approximation of maximum operation to solve the problem in a one-stage paradigm.
- We perform experiments and thus prove the effectiveness of the modeling process and proposed methods.

II. Related Work

Binary Neural Networks (BNNs) aim to represent the activations and weights with 1-bit values. The sign function is widely employed for binarization [5]. However, traditional backward propagation can not work as the derivation result of the sign function is 0. Straight-through Estimator (STE) [8] is employed to approximate the gradients. Based on the basic frameworks, the follow-up methods can be categorized into 1) minimizing the quantization error; 2) improving the loss function; 3) improving the gradient approximation; and 4) designing novel network topology structures.

To minimize the quantization error, Xnor-NET [9] adapt the channel-wise scaling factor and weights. Adabin [7] obtains the optimal binary sets instead of 0/1. For loss function, the key is to add regularization losses or alignment losses. ReActNet [10] designs a standard logit matching loss for attention transfer between BNN and original networks. Meanwhile, ReActNet designs ReAct-Sign and ReAct-PReLU to reshape and shift the activation distributions. LCR [11] retains the Lipschitz constant as a regularization term. For the gradient approximation, FDA [12] decomposes sign the Fourier Series. Nevertheless, these improvements can be combined to better optimize the quantization process. We refer the readers to [13] for more details.

In this report, we learn the adaptive binary sets and model the quantization process into one convex optimization problem. Also, we employ the approximation of maximum operation to make the objective function differential.

Equal Contributions.

III. Model

For the weight $W \in \mathbb{R}^{d_1 \times d_2}$ and binarized weights $W^Q \in \mathbb{R}^{d_1 \times d_2}$ where $W^Q_{i,j} \in \{a, b\}, a \in R, b \in R, a < b, \forall 1 \leq i \leq d_1, 1 \leq j \leq d_2$, the target is to minimize the error under input $X \in \mathbb{R}^{d_2}$:

$$minimize\ \mathcal{L}(WX, W^Q X) \tag{1}$$

We can get two intuitive targets:

$$\mathcal{L}_1(WX, W^Q X) = \sum_{i=1}^{d_1} [(\sum_{j=1}^{d_2} W_{i,j} X_j - \sum_{j=1}^{d_2} W^Q_{i,j} X_j)^2] \tag{2}$$

and

$$\mathcal{L}_2(WX, W^Q X) = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} (W_{i,j} X_j - W^Q_{i,j} X_j)^2 \tag{3}$$

$\mathcal{L}_1$ aims to align the values of the dot product, while $\mathcal{L}_2$ aims to align each product. Theoretically, $\mathcal{L}_1$ equals $\mathcal{L}_2$ plus some covariance items. Since $\mathcal{L}_2$ is easy to **decompose**, we select $\mathcal{L}_2$ as our optimization target in this report.

Therefore, we can get our optimization problem as:

$$\begin{aligned} \underset{a,b}{Min}\, \underset{W^Q}{Min} \ & \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} (W_{i,j} X_j - W^Q_{i,j} X_j)^2 \\ s.t.\ & (W^Q_{i,j} - a)(W^Q_{i,j} - b) = 0 \\ & a \in \mathcal{R}, b \in \mathcal{R} \end{aligned} \tag{4}$$

This problem is a typical constraint optimization problem. Since the formula is **quadratic**, we can easily get that it is **convex** considering W^Q, a, and b.

Consider $(W_{i,j} X_j - W^Q_{i,j} X_j)^2$, we can rewrite it as $X_j^2 (W_{i,j} - W^Q_{i,j})^2$. Therefore, we can get one simple optimal solution:

$$W^{Q*}_{i,j} = \begin{cases} a \text{ if } W_{i,j} < \frac{a+b}{2} \\ b \text{ otherwise} \end{cases} \tag{5}$$

Thus, we get the optimization problem as:

$$\begin{aligned} \underset{a,b}{Min} & \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} (W_{i,j} X_j - W^Q_{i,j} X_j)^2 \\ s.t.\ & W^Q_{i,j} = a \text{ if } W_{i,j} < \frac{a+b}{2} \\ & W^Q_{i,j} = b \text{ if } W_{i,j} \geq \frac{a+b}{2} \\ & a \in \mathcal{R}, b \in \mathcal{R} \end{aligned} \tag{6}$$

However, such a format is not differentiable. Indeed, the target is:

$$\begin{aligned} \underset{a,b}{Min} & \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \min((W_{i,j} - a)^2, (W_{i,j} - b)^2) X_j^2 \\ s.t.\ & a \in \mathcal{R}, b \in \mathcal{R} \end{aligned} \tag{7}$$

Now, we can apply the continuous approximation of the maximum function:

$$\max(x, y) \simeq \frac{\log(\exp(T * x) + \exp(T * y))}{T} \tag{8}$$

where T is a scale factor if $|a - b|$ is small. The larger the gap between $T * a$ and $T * b$, the better the approximation. Therefore, we can get the approximation of the minimum function:

$$\min(x, y) \simeq -\frac{\log(\exp(T * -x) + \exp(T * -y))}{T} \tag{9}$$

Put the Equation 9 into Equation 7, we get final differential unconstrained problem:

$$\begin{aligned} & \underset{a,b}{Min} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} g((W_{i,j} - a)^2, (W_{i,j} - b)^2) X_j^2 \\ & s.t.\ a \in \mathcal{R}, b \in \mathcal{R} \\ & where\ g(x, y) = -\frac{\log(\exp(T * -x) + \exp(T * -y))}{T} \\ & T \in \mathcal{R} \end{aligned} \tag{10}$$

IV. Algorithm

Indeed, the object is

$$\begin{aligned} \underset{a,b}{Min}\, \underset{W^Q}{Min} \ & \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} (W_{i,j} X_j - W^Q_{i,j} X_j)^2 \\ s.t.\ & (W^Q_{i,j} - a)(W^Q_{i,j} - b) = 0 \\ & a \in \mathcal{R}, b \in \mathcal{R} \end{aligned} \tag{11}$$

It is a two-stage optimization:

- step 1: select the $W^Q_{i,j}$ to be a or b to get the minimum.
- step 2: find proper a and b to minimize the minimum in step 1.

For $\mathcal{L}_2$, we can easily find the optimal point after decomposing it. However, for $\mathcal{L}_1$, it is composed and hard to find the optimal point under one specific group of a and b.

For the two-stage object, we then propose a novel Expectation Maximization style method to tackle the problem. The key idea is to alternately perform step 1 and step 2. Specifically, we optimize the strategy to decide $W^Q_{i,j}$ based on initialized a and b. Then we try to find the best a and b under this grouping strategy. After that, we start the recursive process to repeat step 1 to group $W^Q_{i,j}$ under found a and b. The details can be found in 1. Opt denotes the optimization methods such as Newton's descending method.

Algorithm 1: General Two-stage Framework

```
Data: Threshold $\epsilon$, Interface $d$
Result: Optimal $a$ and $b$
$d \leftarrow$ medium of $W$;
while $error \geq \epsilon$ do
    foreach $W^Q_{i,j} \in W^Q$ do
        $W^Q_{i,j} \leftarrow a$ if $W_{i,j} < d$ else $b$ ;
    end
    $a, b, error \leftarrow$ Opt($W^Q, X$) ;
    $d \leftarrow \frac{a+b}{2}$ ;
end
```

In this report, we introduce the approximation of the minimum function to merge two steps. After that, we can **directly optimize** a and b. Therefore, the only thing is to directly apply the optimization method to the object in Equation 10.

V. Evaluation

A. Data

For the test, we select the real value $W \in \mathcal{R}^{10\times 84}$ and feature $X \in \mathcal{R}^{84}$ from the last layer of LeNet-5 [14]. Figure 1 shows the density distribution of W. We can see that the distribution is roughly symmetric around 0.

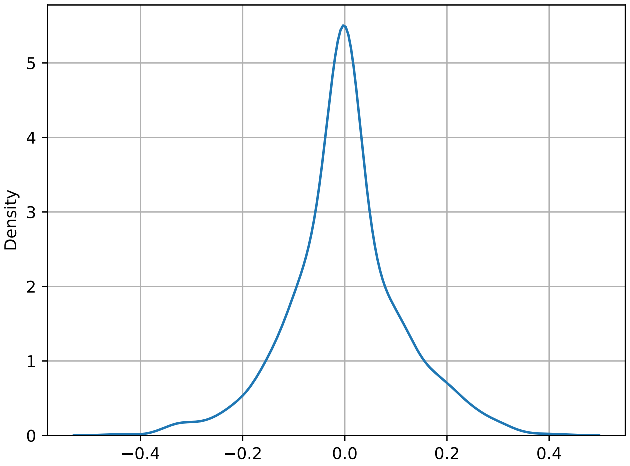

Fig. 1. Density distribution of W.

B. Optimization Methods

In this report, we propose two frameworks to solve the problem.

- one-stage: optimize Equation 10.
- two-stage: optimize Equation 6 following Algorithm 1.

For the optimization methods, we employ the 1) **Newton's method** and 2) **Ellipsoid method**. The gradient under the one-stage framework is:

$$g_{one}(a) = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \frac{2\mathrm{e}^{T\cdot(W_{i,j}-a)^2} X_j^2 \cdot (a - W_{i,j})}{\mathrm{e}^{T\cdot(W_{i,j}-a)^2} + \mathrm{e}^{T\cdot(W_{i,j}-b)^2}} \tag{12}$$

$$g_{one}(b) = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \frac{2\mathrm{e}^{T\cdot(W_{i,j}-b)^2} X_j^2 \cdot (b - W_{i,j})}{\mathrm{e}^{T\cdot(W_{i,j}-a)^2} + \mathrm{e}^{T\cdot(W_{i,j}-b)^2}} \tag{13}$$

The gradient under the two-stage framework is:

$$g_{two}(a) = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} -2*(W_{i,j}X_j - aX_j)X_j \cdot I(W_{i,j} < \frac{a+b}{2}) \tag{14}$$

$$g_{two}(b) = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} -2*(W_{i,j}X_j - bX_j)X_j \cdot I(W_{i,j} \geq \frac{a+b}{2}) \tag{15}$$

where the I denotes the indicator function.

C. Main Results

During testing, we set the threshold ϵ as 1e-6 and train 10000 interactions with early-stop. For one-stage algorithms, the T is 100.

Ellipsoid Method. We visualize the ellipsoid for the first 25 interactions and the coverage points. As shown in Figure 2 and Figure 3, we can see that both one-stage and two-stage algorithms coverage.

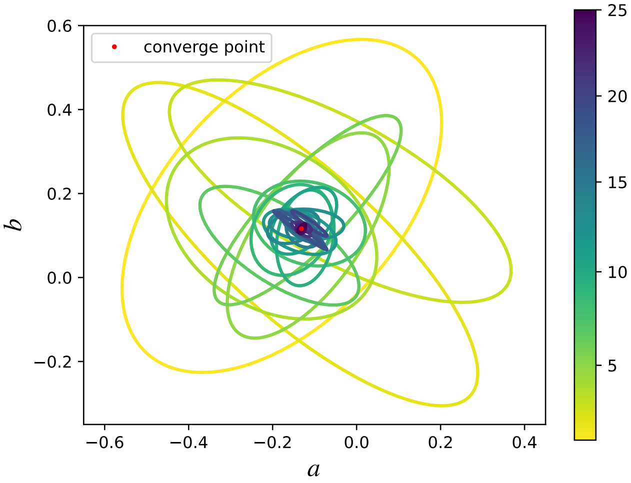

Fig. 2. First 25 ellipsoids and coverage points for one-stage algorithm.

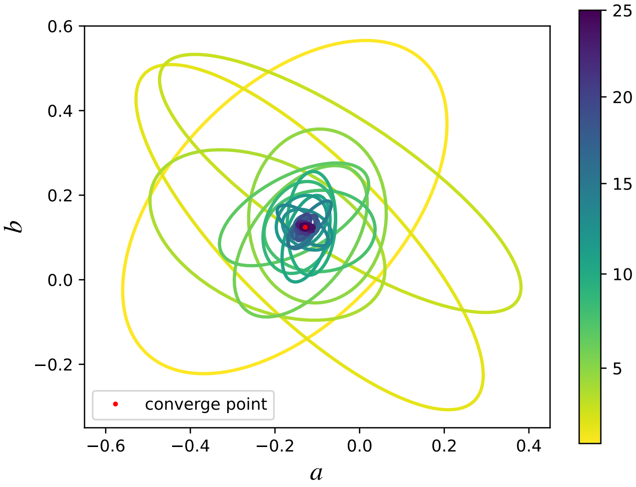

Fig. 3. First 25 ellipsoids and coverage points for the two-stage algorithm.

Convergence. We plot the losses during the training process in Figure 4. The conclusions are as follows:

- Both one-stage methods and two-stage methods outperform the Adabin, indicating the effectiveness of the modeling process. Meanwhile, both Newton's method and the Ellipsoid method converge well and get better results than Adabin.
- For both Newton's method and the Ellipsoid method, the one-stage framework converges to better results. It shows the effectiveness of the proposed approximation.

- Newton's method coverage faster than the Ellipsoid method for both the one-stage framework and the two-stage framework.
- For the Ellipsoid method, the loss sometimes gets larger. It consists of the conclusion that the Ellipsoid method is not a descent method.

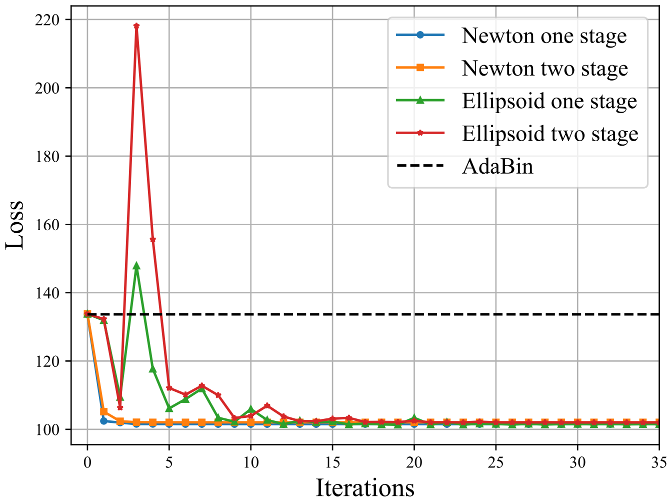

Fig. 4. The training losses for baseline and proposed methods.

Optimal a and b. We list the optimal a, b and corresponding points binarized to a or b. As shown in Table I, we can observe that Adabin would get a symmetric group of a and b, and thus the interface is around 0.001. Both one-stage and two-stage methods get a left shift for the interface. Meanwhile, for the one-stage framework, Newton's method and the Ellipsoid method converge to the same optimal point. However, for the two-stage framework, they converge to the different optimal points as the coupling of the two stages may mislead the learning process to the local minima.

TABLE I
THE CONVERGED a AND b FOR BASELINES. #P MEANS THE NUMBER OF POINTS WHICH ARE BINARIZED TO a OR b. I DENOTES THE ONE-STAGE METHOD AND II DENOTES THE TWO-STAGE METHOD.

Method	a	b	Interface	#P to a	#P to b
Adabin [7]	-0.0771	0.0774	-0.0008	420	420
Newton I	-0.1312	0.1161	-0.0075	360	480
Newton II	-0.1276	0.1248	-0.0014	411	429
Ellipsoid I	-0.1312	0.1161	-0.0075	360	480
Ellipsoid II	-0.1284	0.1241	-0.0021	408	432

D. Analysis

Impact of T. To study the impact of T on the one-stage algorithms, we change T from 50 to 200. The results are shown in Figure 5. We can conclude that:

- As the T increases, the losses would be lower for both Newton's method and the Ellipsoid method. In the approximation, a larger T would lead to a smaller gap.
- When T increases, the Ellipsoid method would get better results, while Newton's method gets worse at $T = 300$. This indicates that the Ellipsoid method is more robust.

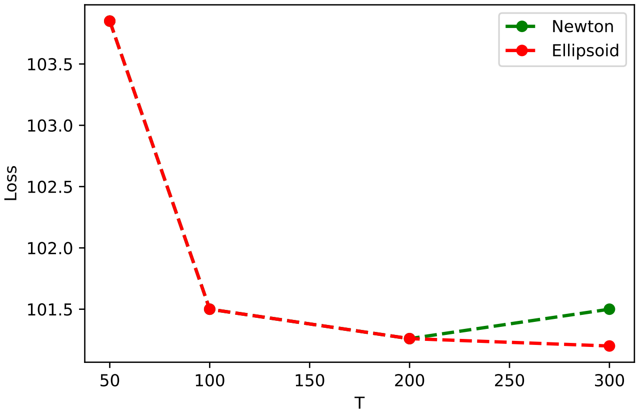

Fig. 5. The coverage losses for one-stage Newton's method and one-stage Ellipsoid method under different T.

Robustness towards noisy gradient. To study the robustness towards noisy gradients, we add some Gaussian noise to the gradients. As shown in Table II, we can conclude that Newton's method is more robust to the noise and the Ellipsoid method can not converge with larger noise. Meanwhile, the one-stage method is more robust than the two-stage method.

TABLE II
PERFORMANCE OF THE PROPOSED METHOD WHEN ADDING VARIOUS NOISES TO THE GRADIENTS. FOR THE NOISE, THE VARIANCE CHANGES FROM 0.1 TO 0.5. I DENOTES THE ONE-STAGE METHOD AND II DENOTES THE TWO-STAGE METHOD. NAN MEANS THAT THE METHOD IS DIVERGENT.

Method	0	0.01	0.1	0.5	1.0	1.3
Newton I	101.5	101.5	101.5	101.5	101.5	101.5
Newton II	102.0	102.0	102.0	102.0	102.0	NaN
Ellipsoid I	101.5	101.5	101.5	101.5	NaN	NaN
Ellipsoid II	102.0	102.0	102.0	NaN	NaN	NaN

VI. CONCLUSION

In this report, we model the binarizing process as a convex problem and design a two-stage algorithm to solve this problem. Furthermore, we employ the approximation of maximum function and convey it into one differential unconstrained problem. Therefore, we propose a novel one-stage framework. Experimental results prove the effectiveness of the modeling process and proposed methods.

For future work, we can consider the binarizing process of activation functions. Also, how to guarantee the performance of more than one layer is an interesting topic.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[4] C. Ko, Z. Lyu, L. Weng, L. Daniel, N. Wong, and D. Lin, "POPQORN: quantifying robustness of recurrent neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 3468–3477, PMLR, 2019.

[5] M. Courbariaux and Y. Bengio, "Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1," *CoRR*, vol. abs/1602.02830, 2016.

[6] H. Bai, W. Zhang, L. Hou, L. Shang, J. Jin, X. Jiang, Q. Liu, M. R. Lyu, and I. King, "Binarybert: Pushing the limit of BERT quantization," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021* (C. Zong, F. Xia, W. Li, and R. Navigli, eds.), pp. 4334–4348, Association for Computational Linguistics, 2021.

[7] Z. Tu, X. Chen, P. Ren, and Y. Wang, "Adabin: Improving binary neural networks with adaptive binary sets," in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XI* (S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), vol. 13671 of *Lecture Notes in Computer Science*, pp. 379–395, Springer, 2022.

[8] Y. Bengio, N. Léonard, and A. C. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *CoRR*, vol. abs/1308.3432, 2013.

[9] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), vol. 9908 of *Lecture Notes in Computer Science*, pp. 525–542, Springer, 2016.

[10] Z. Liu, Z. Shen, M. Savvides, and K. Cheng, "Reactnet: Towards precise binary neural network with generalized activation functions," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV* (A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, eds.), vol. 12359 of *Lecture Notes in Computer Science*, pp. 143–159, Springer, 2020.

[11] Y. Shang, D. Xu, B. Duan, Z. Zong, L. Nie, and Y. Yan, "Lipschitz continuity retained binary neural network," in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XI* (S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), vol. 13671 of *Lecture Notes in Computer Science*, pp. 603–619, Springer, 2022.

[12] Y. Xu, K. Han, C. Xu, Y. Tang, C. Xu, and Y. Wang, "Learning frequency domain approximation for binary neural networks," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual* (M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, eds.), pp. 25553–25565, 2021.

[13] C. Yuan and S. S. Agaian, "A comprehensive review of binary neural network," *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 12949–13013, 2023.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.